UCEpic : Unifying Aspect Planning and Lexical Constraints for Generating Explanation Recommendation

Advisor :Jia-Ling, Koh

Speaker : Shu-Ming Yu

Source : KDD’ 23

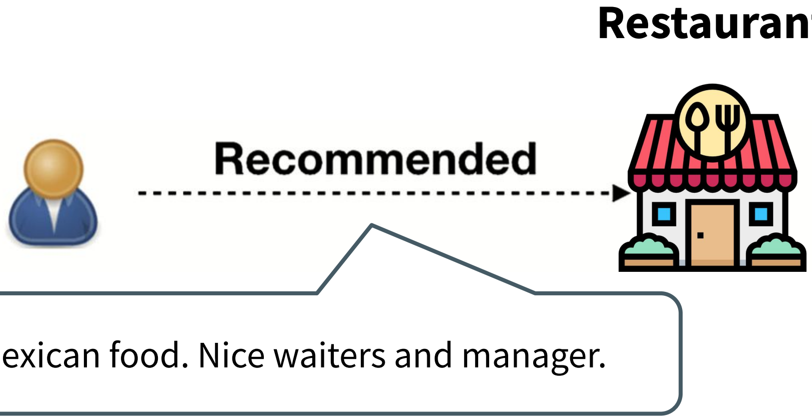
Date : 2023/09/26

Outline

- **Introduction**
- Method
- Experiment
- Conclusion

Explanation Recommendation

- Generating reasonable sentences as explanations of recommended items for users
- The generator is based on natural language generation models



Input / Output

Input:

- Historical review profile of **user** u and **item** i
  - R^u, R^i
- Aspects extract from review for **user** u and **item** i
  - A^{ui}
- Lexical constraints (e.g., keywords) for **user** u and **item** i
  - C^{ui}

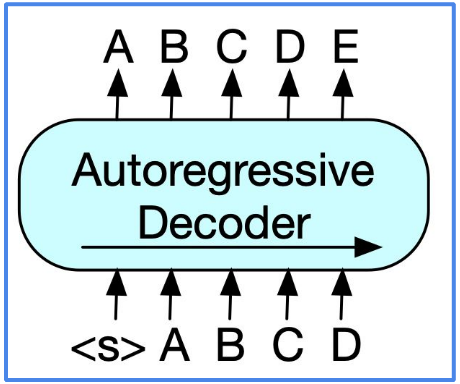
Output:

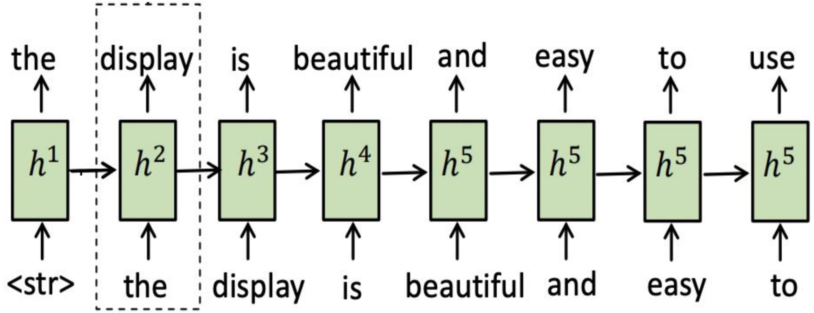
- Generated explanation of **user** u to **item** i
  - E^{ui}

Generation framework

- Auto-regressive generation

GPT

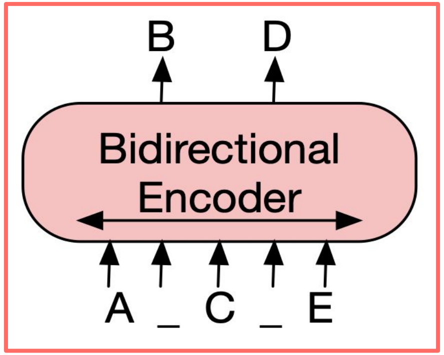




Generation framework

- Insertion-based generation

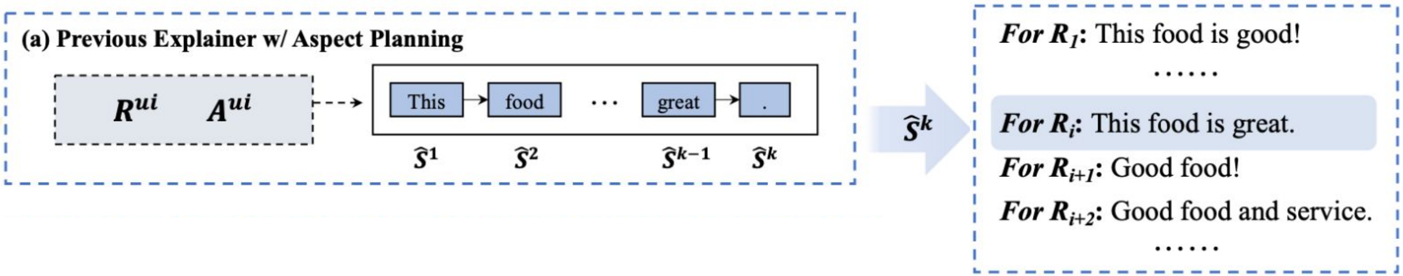
BERT



Stage	Generated text sequence
0 (X^0)	pepper chicken
1 (X^1)	pepper sauce chicken
2 (X^2)	spicy pepper sauce chicken

Aspect planning

- Aspects (e.g., **display** for a TV) mostly control the high-level sentiment
- Disadvantage :
  - Generating too general sentences (e.g., "good screen!")
  - Generating with inaccurate details (e.g., "2K screen" for a 4K TV)



Lexical constraint

- Requiring the generated sentence contain the lexical constraints (e.g., keywords)
- Disadvantage :
  - Model tends to generate similar text
  - Struggle to include specific information in explanation

Keyword : ‘pepper chicken’

Stage	Generated text sequence
$0\,(X^0)$	pepper chicken
$1\,(X^1)$	pepper sauce chicken
$2\,(X^2)$	spicy pepper sauce chicken

Outline

- Introduction
- **Method**
- Experiment
- Conclusion

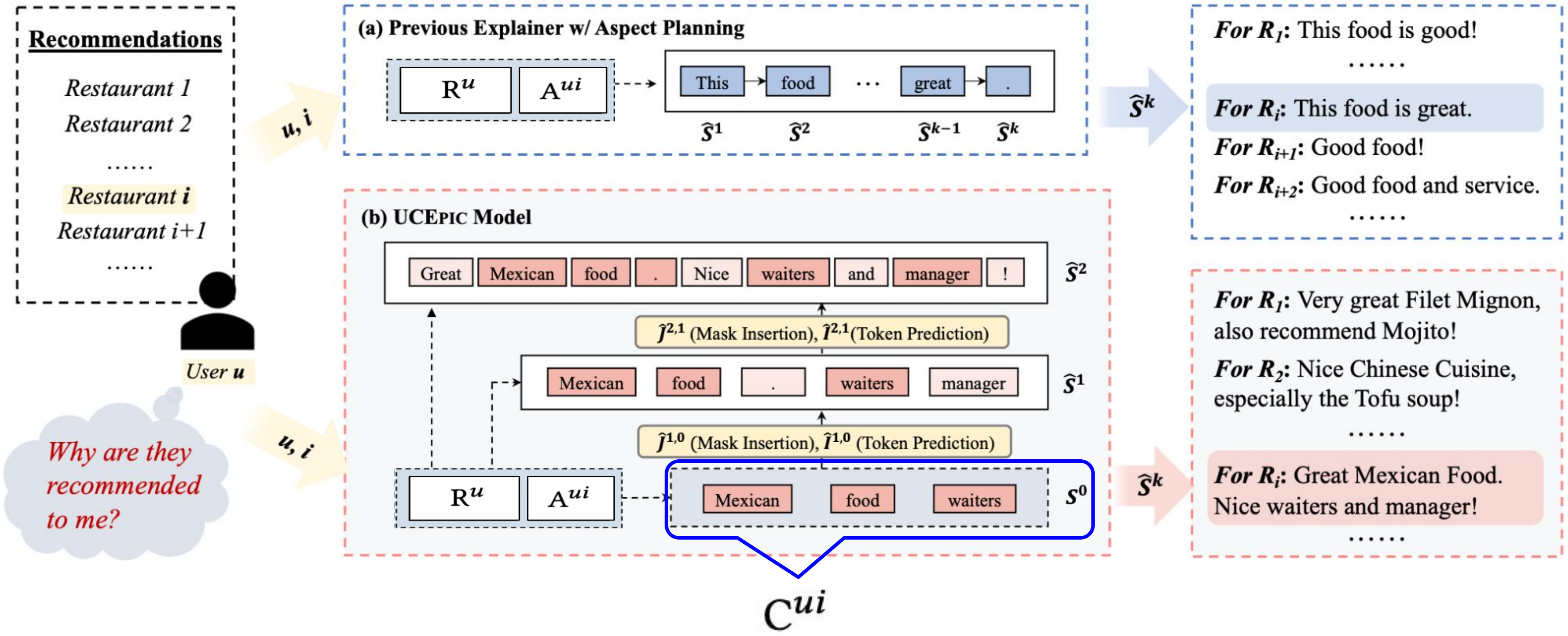
Input / Output

Input:

- Historical review profile of **user** u and **item** i
  - $\mathrm{R}^u, \mathrm{R}^i$
- Aspects extract from review for **user** u and **item** i
  - A^{ui}
- Lexical constraints (e.g., keywords) for **user** u and **item** i
  - C^{ui}

Output:

- Generated explanation of **user** u to **item** i
  - E^{ui}


Recommendations
Restaurant 1
Restaurant 2
......
Restaurant i
Restaurant i+1
......
User u
Why are they recommended to me?
u, i
(a) Previous Explainer w/ Aspect Planning
R^u A^{ui}
This food ... great .
$\hat{S}^1$ $\hat{S}^2$ $\hat{S}^{k-1}$ $\hat{S}^k$
(b) UCEPIC Model
Great Mexican food . Nice waiters and manager !
$\hat{S}^2$
$\hat{J}^{2,1}$ (Mask Insertion), $\hat{I}^{2,1}$(Token Prediction)
Mexican food . waiters manager
$\hat{S}^1$
$\hat{J}^{1,0}$ (Mask Insertion), $\hat{I}^{1,0}$ (Token Prediction)
Mexican food waiters
S^0
C^{ui}
For R_1: This food is good!
......
For R_i: This food is great.
For R_{i+1}: Good food!
For R_{i+2}: Good food and service.
......
For R_1: Very great Filet Mignon, also recommend Mojito!
For R_2: Nice Chinese Cuisine, especially the Tofu soup!
......
For R_i: Great Mexican Food. Nice waiters and manager!
......

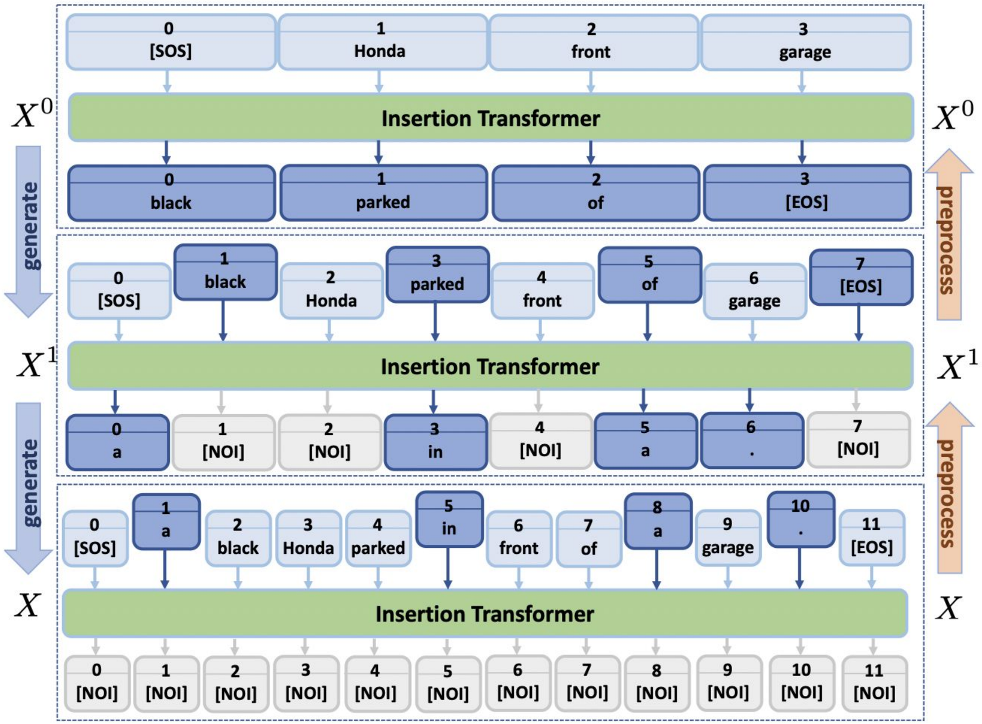
Training step

- Pre-train
  - data construction
  - training

- Fine-tune
  - Only aspect planning
  - Lexical constraints

Data construction

- Insertion-based generation
  - Generate
  - Preprocess



Data construction

- S^K : original sentence
- $I^{K,K-1}$: random **mask** some token from S^K by p ← $\boxed{0.2}$
- $J^{K,K-1}$: recording the mask position and length
- S^{K-1} : masked sentence

…
- S^0 : lexical constraints

Data construction

S(k)	<s>	what	a	cute	baby	</s>
I(k, k-1)	<s>	what	[mask]	[mask]	baby	</s>
J(k, k-1)	0	2	0			
S(k-1)	<s>	what	baby	</s>		
…						
S(0)	<s>	baby	</s>			

$$(S^{k-1}, I^{k,k-1}, J^{k,k-1}, S^k)$$

Pre-training

- Input: $(\hat{S}^{k-1}, \hat{I}^{k,k-1}, \hat{J}^{k,k-1}, \hat{S}^{k})$
- **Learning** how to generate S^K from S^{K-1}

Algorithm 1 **Insertion in the k-th Stage**

---

procedure INSERTION($\hat{S}^{k-1}$)
  $\hat{J}^{k,k-1} \leftarrow$ predict number of masks from $\hat{S}^{k-1}$ via eq. (1) ;
  $\hat{I}^{k,k-1} \leftarrow$ build intermediate sequence from $\hat{J}^{k,k-1}$ and $\hat{S}^{k-1}$;
  $\hat{S}^{k} \leftarrow$ predict masked tokens in $\hat{I}^{k,k-1}$ via eq. (2);
  return predicted sequence $\hat{S}^{k}$;

Pre-training

* MI : mask insertion
TP : token prediction

- Input: $(\hat{S}^{k-1}, \hat{I}^{k,k-1}, \hat{J}^{k,k-1}, \hat{S}^{k})$

Linear projection

$$y_{MI} = \boxed{\mathbf{H}_{MI}}(\boxed{\mathbf{D}}(\hat{S}^{k-1})),\ \hat{J}^{k,k-1} = \mathrm{argmax}(y_{MI})\ ,\quad y_{MI} \in \mathbb{R}^{\boxed{l_s} \times \boxed{d_{ins}}}$$

Max number of insert
= (1/(1-p)) * len($\hat{S}^{K-1}$)

len($\hat{S}^{K-1}$)

=> decided **how much** to insert

MLP bi-directional transformer

$$y_{TP} = \boxed{\mathbf{H}_{TP}}(\boxed{\mathbf{D}}(\hat{I}^{k,k-1})),\ \hat{S}^{k} = \mathrm{argmax}(y_{TP})\ ,\quad y_{TP} \in \mathbb{R}^{\boxed{l_I} \times \boxed{d_{vocab}}}$$

Size of vocab

len($\hat{I}^{K,K-1}$)

=> decided **what words** to insert

Fine-tune

- Input: $(S_+^{k-1}, I_+^{k,k-1}, J_+^{k,k-1}, S_+^k)$
- Fine-tune the model **with personalized references and aspect information**.

$$S_+^k = [\mathrm{R}^{ui}, \mathrm{A}^{ui}, S^k]$$
$$= [w_0^r, \ldots, w_{|\mathrm{R}^{ui}|}^r, w_0^a, \ldots, w_{|\mathrm{A}^{ui}|}^a, w_0, \ldots, w_{|S^k|}]$$

$$J_+^{k,k-1} = [\mathbf{0}_{|\mathrm{R}^{ui}|}, \mathbf{0}_{|\mathrm{A}^{ui}|}, J^{k,k-1}]$$

$$I_+^{k,k-1} = [\mathrm{R}^{ui}, \mathrm{A}^{ui}, I^{k,k-1}]$$

Fine-tune

- Input: $(S_{+}^{k-1}, I_{+}^{k,k-1}, J_{+}^{k,k-1}, S_{+}^{k})$

Input S^K :	**pre-train**		**fine-tune**
	$\hat{S}^k = [S^k]$	$\longrightarrow$	$S_+^k = [\mathrm{R}^{ui}, \mathrm{A}^{ui}, S^k]$

$$[\mathrm{O}_S^{R^{ui}}, \mathrm{O}_S^{A^{ui}}, \mathrm{O}^{S^k}] = \mathbf{D}(\hat{S}_+^k) \qquad y_{MI} = \mathbf{H}_{MI}(\mathrm{O}^{S^k})$$

$$[\mathrm{O}_I^{R^{ui}}, \mathrm{O}_I^{A^{ui}}, \mathrm{O}^{I^{k,k-1}}] = \mathbf{D}(\hat{I}_+^{k,k-1}) \qquad y_{TP} = \mathbf{H}_{TP}(\mathrm{O}^{I^{k,k-1}})$$

Fine-tune

- Input: $(S_{+}^{k-1}, I_{+}^{k,k-1}, J_{+}^{k,k-1}, S_{+}^{k})$
  - aspect starting stage (no existing tokens)

    $$S_{+a}^{0} = [\mathrm{R}^{ui}, \mathrm{A}^{ui}]$$

  - lexical constraint starting stage

    $$S_{+l}^{0} = [\mathrm{R}^{ui}, \mathrm{A}^{pad}, \mathrm{C}^{ui}]$$

special aspect for lexical constraints

Loss

$$\mathcal{L} = -\log p(S_+^k | S_+^{k-1})$$

$$= -\log \underbrace{p(S_+^k | I_+^{k,k-1})}_{\text{Token prediction}} \underbrace{p(J_+^{k,k-1} | S_+^{k-1})}_{\text{Mask insertion}},$$

Outline

- Introduction
- Method
- **Experiment**
- Conclusion

Experiment

- **Dataset**

  Use wikipedia for pre-training

  Fine-tune on 1. **RateBeer** : beer reviews from ratebeer

  2. **Yelp** : restaurant reviews on Yelp

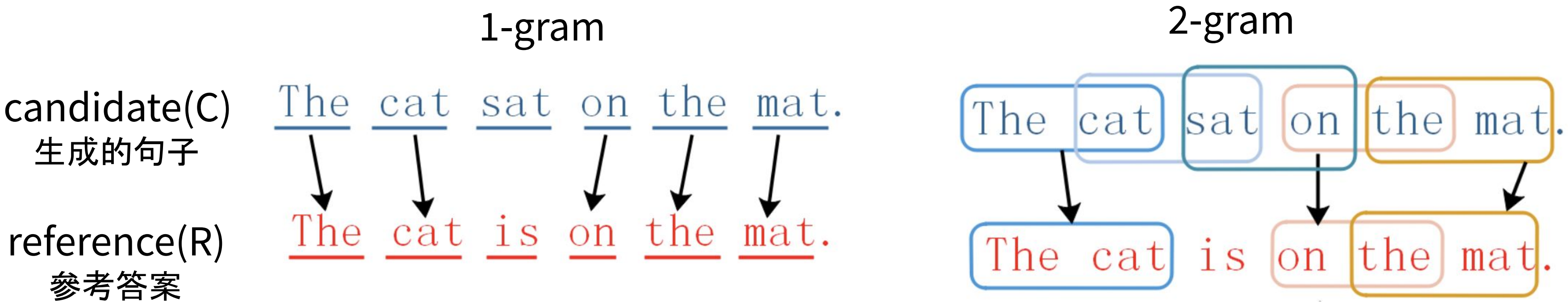
Dataset	Train	Dev	Test	#Users	#Items	#Aspects
RateBeer	16,839	1,473	912	4,385	6,183	8
Yelp	252,087	37,662	12,426	235,794	22,412	59

Experiment

Candidate : 生成的句子

Reference : 參考答案

- **N-gram**



Candidate : 生成的句子

Reference : 參考答案

Experiment

- **Evaluation**
  - BLEU

$$BLEU = BP \cdot exp(\sum_{n=1}^{N} w_n \log p_n)$$

$$p_n = \frac{\sum_{C \in \{Candidates\}} \sum_{n\text{-}gram \in C} Count_{clip}(n\text{-}gram)}{\boxed{\sum_{C' \in \{Candidates\}} \sum_{n\text{-}gram' \in C'} Count(n\text{-}gram')}}.$$

Candidate中n-gram有幾個

candidate(C)
生成的句子

1-gram

The cat sat on the mat.

6 個

2-gram

The cat sat on the mat.

5 個

Experiment

Candidate : 生成的句子

Reference : 參考答案

- **Evaluation**
  - BLEU

$$BLEU = BP \cdot exp(\sum_{n=1}^{N} w_n log\, p_n)$$

$$p_n = \frac{\sum_{C \in \{Candidates\}} \sum_{n\text{-}gram \in C} Count_{clip}(n\text{-}gram)}{\sum_{C' \in \{Candidates\}} \sum_{n\text{-}gram' \in C'} Count(n\text{-}gram')}.$$

$$Count_{clip} = \min(\underline{Count}, \underline{Max\_Ref\_Count})$$

candidate中這個word出現的次數

reference 中這個word出現最多的次數

1-gram

candidate(C) The $\boxed{\text{cat}}$ sat on the mat.

reference(R) The $\boxed{\text{cat}}$ is on the mat.

$Count_{clip}$ = min(1, max(1))

= 1

Candidate : 生成的句子

Reference : 參考答案

Experiment

- **Evaluation**
  - BLEU

$$BLEU = BP \cdot exp(\sum_{n=1}^{N} w_n log\, p_n)$$

$$p_n = \frac{\sum_{C\in\{Candidates\}} \sum_{n\text{-}gram\in C} Count_{clip}(n\text{-}gram)}{\sum_{C'\in\{Candidates\}} \sum_{n\text{-}gram'\in C'} Count(n\text{-}gram')}.$$

Candidate: the the the the the the the. 7 個the

Reference 1: The cat is on the mat. 2 個the

$$Count_{clip} = \min(\underline{Count}, \underline{Max\_Ref\_Count})$$

candidate中這個word出現的次數

reference 中這個word出現最多的次數

$Count_{clip}$ = min(7, max(2))
= 2

Experiment

Candidate : 生成的句子

Reference : 參考答案

- **Evaluation**
  - BLEU

$$BLEU = BP \cdot exp(\sum_{n=1}^{N} w_n log\, p_n)$$

$$p_n = \frac{\sum_{C \in \{Candidates\}} \sum_{n\text{-}gram \in C} Count_{clip}(n\text{-}gram)}{\sum_{C' \in \{Candidates\}} \sum_{n\text{-}gram' \in C'} Count(n\text{-}gram')}.$$

$$Count_{clip} = \min(\underline{Count}, \underline{Max\_Ref\_Count})$$

candidate中這個word出現的次數

reference 中這個word出現最多的次數

Candidate: the the the the the the the. 7 個the

Reference 1: The cat is on the mat. 2 個the

Reference 2: There is a cat on the mat. 1 個the

$Count_{clip}$ = min(7, max(2, 1))
= 2

Experiment

Candidate : 生成的句子

Reference : 參考答案

- **Evaluation**
  - BLEU

$$BLEU = BP \cdot exp(\sum_{n=1}^{N} w_n \log p_n)$$

$$p_n = \frac{\sum_{C \in \{Candidates\}} \sum_{n\text{-}gram \in C} Count_{clip}(n\text{-}gram)}{\sum_{C' \in \{Candidates\}} \sum_{n\text{-}gram' \in C'} Count(n\text{-}gram')}.$$

1-gram

candidate(C) 生成的句子: The cat sat on the mat.

reference(R) 參考答案: The cat is on the mat.

P1 = 5/6

$$BP = \begin{cases} 1 & lc > lr \\ \exp(1 - lr/lc) & lc \leq lr \end{cases}$$

lc = 机器译文的长度

lr = 最短的参考翻译句子的长度

Candidate : 生成的句子

Experiment

Reference : 參考答案

- **Evaluation**
  - BLEU as **precision**

$$BLEU = BP \cdot exp(\sum_{n=1}^{N} w_n log\, p_n)$$

$$p_n = \frac{\sum_{C \in \{Candidates\}} \sum_{n\text{-}gram \in C} Count_{clip}(n\text{-}gram)}{\sum_{C' \in \{Candidates\}} \sum_{n\text{-}gram' \in C'} Count(n\text{-}gram')}.$$

1-gram

candidate(C) 生成的句子 The cat sat on the mat.

reference(R) 參考答案 The cat is on the mat.

P1 = 5/6

Experiment

Candidate : 生成的句子

Reference : 參考答案

- **Evaluation**
  - ROUGE-N

$$ROUGE-N = \frac{\sum_{S \in \{ReferenceSummaries\}} \sum_{gram_n \in S} Count_{match}(gram_n)}{\sum_{S \in \{ReferenceSummaries\}} \sum_{gram_n \in S} Count(gram_n)}$$

$$p_n = \frac{\sum_{C \in \{Candidates\}} \sum_{n\text{-}gram \in C} Count_{clip}(n\text{-}gram)}{\sum_{C' \in \{Candidates\}} \sum_{n\text{-}gram' \in C'} Count(n\text{-}gram')}.$$

1-gram

candidate(C) 生成的句子: The cat sat on the mat.

reference(R) 參考答案: The cat is on the mat.

ROUGE-1 = 5/6

Candidate : 生成的句子

Reference : 參考答案

Experiment

- **Evaluation**
  - ROUGE-L

$$ROUGE-L = \frac{LCS(X,Y)}{m}$$

LCS : longest common subsequence

m : len(reference)

1-gram

candidate(C) 生成的句子	The cat sat on the mat.
reference(R) 參考答案	The cat is on the mat.

ROUGE-L = 5/6

Candidate : 生成的句子

Experiment

Reference : 參考答案

- **Evaluation**
  - BLEU as **precision**
  - ROUGE as **recall**

Candidate : 生成的句子

Reference : 參考答案

Experiment

- **Evaluation**
  - BLEU as **precision**
  - ROUGE as **recall**
  - METEOR

$$METEOR = (1 - pen) \times F_{means}$$

$$F_{means} = \frac{PR}{\alpha P + (1-\alpha) R}$$

α = 0.5

=> F_{means} as F-1

P : precision
R : recall

Experiment

Candidate : 生成的句子

Reference : 參考答案

• Evaluation

- BLEU as **precision**
- ROUGE as **recall**
- METEOR as **F-1**

$$METEOR = (1 - pen) \times F_{means}$$

$$F_{means} = \frac{PR}{\alpha P + (1-\alpha)R}$$

$$Pen = \frac{\#chunks}{m}$$

m : number of match

1-gram

candidate(C)
生成的句子

$\boxed{\text{The cat}}$ sat $\boxed{\text{on the mat.}}$

reference(R)
參考答案

$\boxed{\text{The cat}}$ is $\boxed{\text{on the mat.}}$

Experiment

- **Evaluation**
  - BLEU as **precision**
  - ROUGE as **recall**
  - METEOR as **F-1**
  - Distinct

$$\text{Distinct-n} : \frac{\text{count(unique gram)}}{\text{len(candidate)}}$$

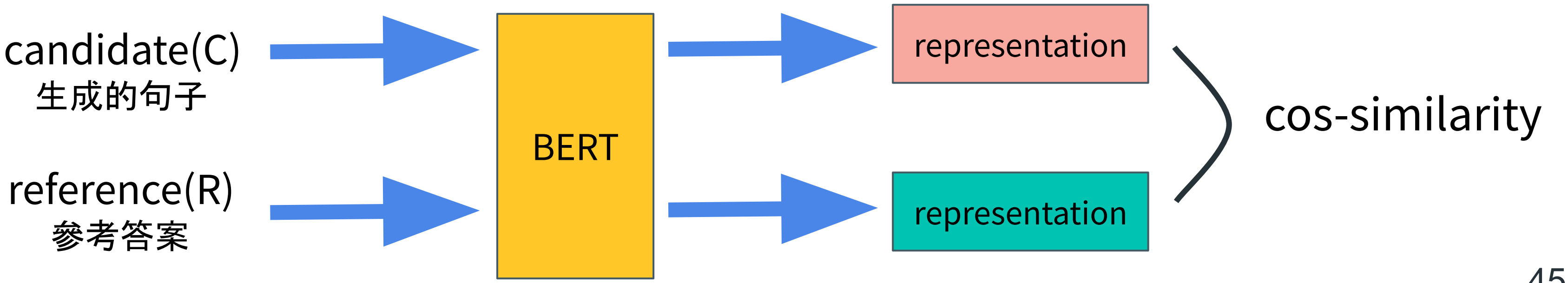
1-gram

candidate(C) The cat sat on the mat.
生成的句子

Distinct-1 = 5/6

Experiment

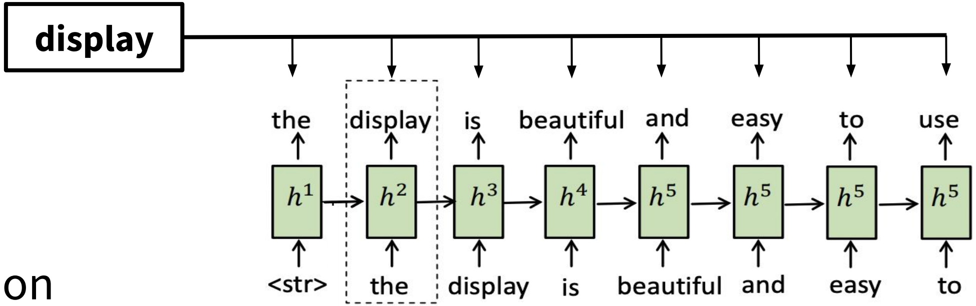
- **Evaluation**
  - BLEU as **precision**
  - ROUGE as **recall**
  - METEOR as **F-1**
  - Distinct
  - BERT-score



Experiment

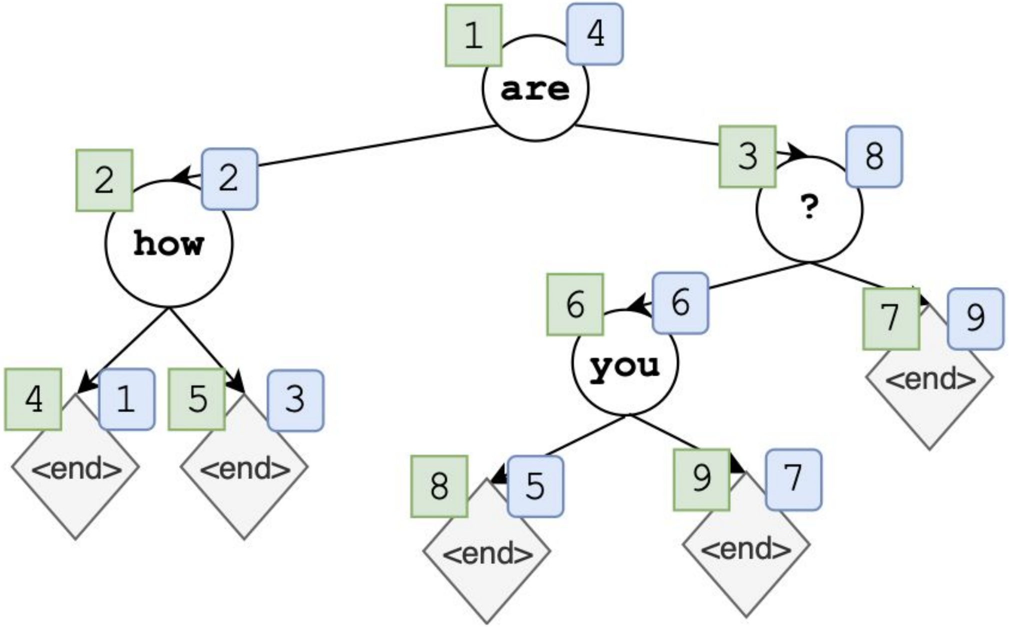
- **Baseline**
  - ExpansionNet
  - Ref2Seq
  - PETER
  - ↑ Auto-regressive generation

  - ↓ Insertion-based generation
  - NMSTG
  - POINTER
  - CBART



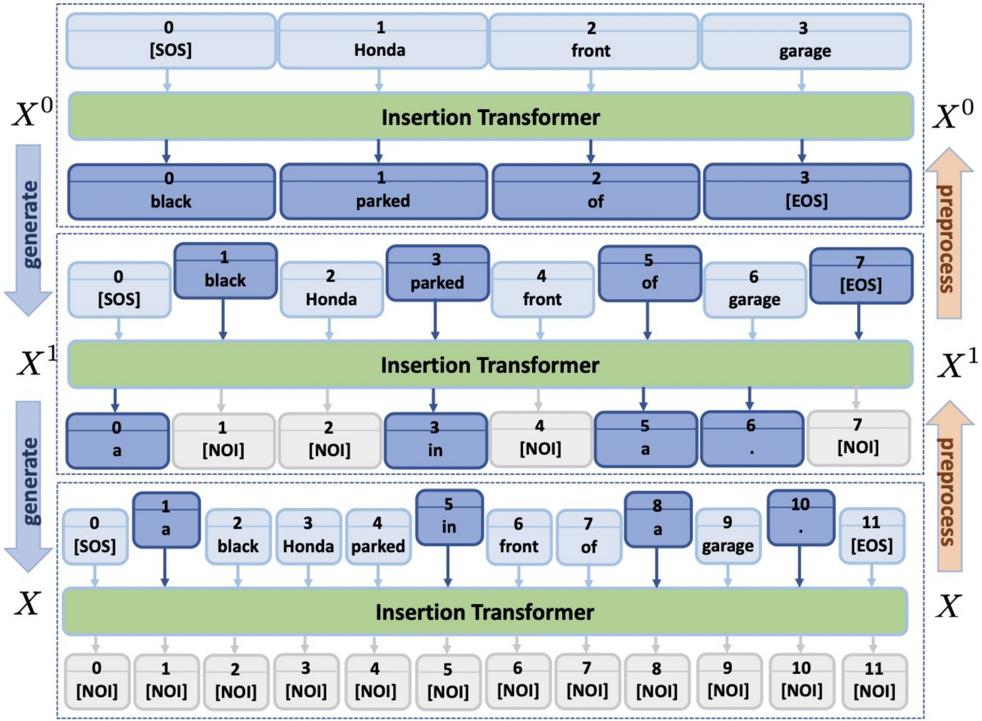
Experiment

- **Baseline**
  - NMSTG



Experiment

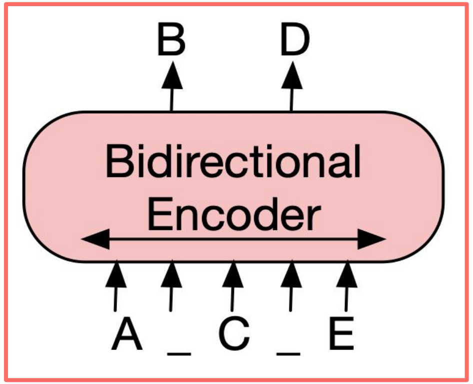
- **Baseline**
  - POINTER



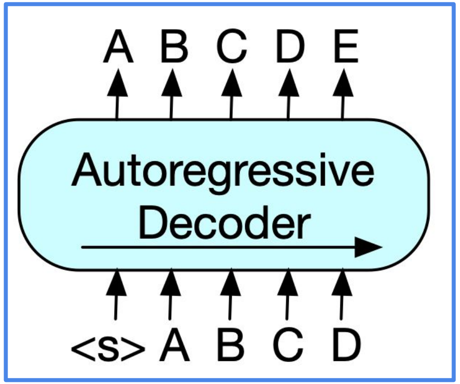
Experiment

- **Baseline**
  - CBART

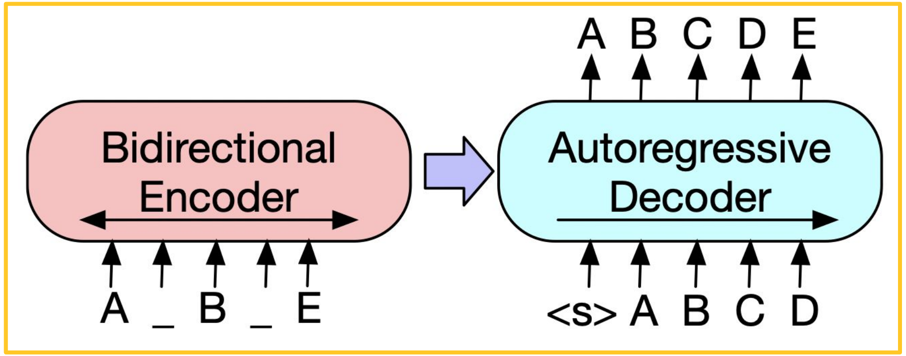
BERT



GPT



BART



Experiment

- Lexically constrain

Models	RateBeer							Yelp						
	B-1	B-2	D-1	D-2	M	R	BS	B-1	B-2	D-1	D-2	M	R	BS
Human-Oracle	–	–	8.30	49.16	–	–	–	–	–	3.8	34.1	–	–	–
Lexically constrained generation														
ExpansionNet	5.41	0.49	0.97	4.91	6.09	5.55	76.14	1.49	0.08	0.40	1.90	2.19	1.93	73.68
Ref2Seq	17.94	4.50	1.09	5.49	17.03	15.17	83.72	6.38	0.77	0.51	3.64	7.02	10.58	82.88
PETER	15.03	2.46	2.04	11.40	9.49	13.27	79.08	7.59	1.32	1.52	8.70	7.64	12.24	80.89
NMSTG	22.82	2.30	6.02	50.39	15.17	15.35	82.31	13.67	0.77	**4.57**	**57.02**	9.64	11.13	80.80
POINTER	6.00	0.31	**11.24**	**56.02**	7.41	11.21	81.80	1.50	0.06	5.49	29.76	3.24	5.23	80.85
CBART	2.49	0.54	8.49	34.74	8.45	13.84	83.30	2.19	0.60	5.32	26.79	9.41	15.00	84.08
UCEPIC	**27.97**	**5.09**	5.24	32.04	**19.90**	**17.05**	**84.03**	**13.77**	**3.06**	2.85	20.39	**14.45**	**16.92**	**84.55**

Experiment

- aspect-planning v.s. lexically constrain

Models	RateBeer							Yelp						
	B-1	B-2	D-1	D-2	M	R	BS	B-1	B-2	D-1	D-2	M	R	BS
Human-Oracle	–	–	8.30	49.16	–	–	–	–	–	3.8	34.1	–	–	–
Aspect-planning generation														
ExpansionNet	8.96	1.79	0.20	1.05	16.30	10.13	75.58	4.92	0.47	0.18	1.40	7.78	5.42	76.27
Ref2Seq	17.15	4.17	0.95	4.41	16.66	15.66	80.76	8.34	0.98	0.46	3.77	7.58	11.19	82.66
PETER	25.25	5.35	0.74	3.44	19.19	20.34	84.03	14.26	2.25	0.26	1.23	12.25	14.75	82.55
UCEpic	27.42	2.89	4.49	29.23	19.54	15.48	83.53	8.03	0.72	1.89	14.75	8.10	11.58	83.53
Lexically constrained generation														
ExpansionNet	5.41	0.49	0.97	4.91	6.09	5.55	76.14	1.49	0.08	0.40	1.90	2.19	1.93	73.68
Ref2Seq	17.94	4.50	1.09	5.49	17.03	15.17	83.72	6.38	0.77	0.51	3.64	7.02	10.58	82.88
PETER	15.03	2.46	2.04	11.40	9.49	13.27	79.08	7.59	1.32	1.52	8.70	7.64	12.24	80.89
UCEpic	**27.97**	**5.09**	5.24	32.04	**19.90**	**17.05**	**84.03**	**13.77**	**3.06**	2.85	20.39	**14.45**	**16.92**	**84.55**

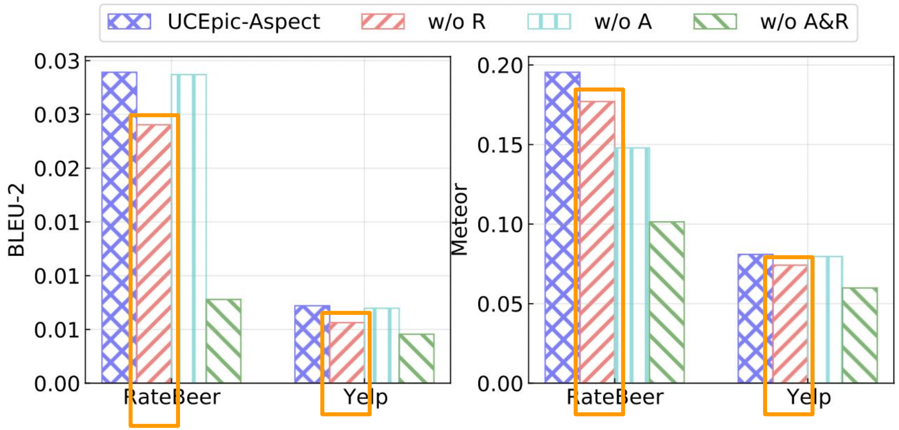
Experiment

	Phrases	pepper chicken	north shore , meat
	Human	Food was great. The pepper chicken is the best. This place is neat and clean. The staff are sweet. I recomend them to anyone!!	Great Italian food on the north shore ! Menu changes daily based on the ingredients they can get locally. Everything is organic and made "clean". There is no freezer on the property, so you know the meat was caught or prepared that day. The chef is also from Italy! I highly recommend!
Auto-regression	Ref2Seq	best restaurant in town ! ! !	what a good place to eat in the middle of the area . the food was good and the service was good .
	PETER	This place is great! I love the food and the service is always great. I love the chicken and the chicken fried rice. I love this place.	The food was good, but the service was terrible. The kitchen was not very busy and the kitchen was not busy. The kitchen was very busy and the kitchen was not busy.
Insertion-based	POINTER	pepper sauce chicken !	one of the best restaurants in the north as far as i love the south shore . great meat ! !
	CBART	Great spicy pepper buffalo wings and chicken wings.	Best pizza on the north shore ever! Meatloaf is to die for, especially with meat lovers.
	UCEPIC	Great Chinese restaurant, really great food! The customer service are amazing! Everything is delicious and delicious! I think this local red hot pepper chicken is the best.	I had the best Italian north shore food. The service is great, meat that is fresh and delicious. Highly recommend!

Experiment

- Reviews' information is very important for generation

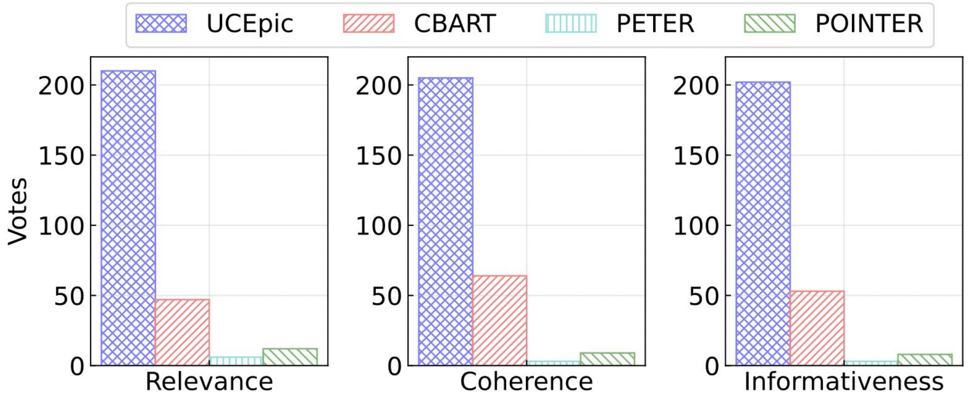
R : $\mathbf{R}^{u}$
A : $\mathbf{A}^{ui}$



Experiment

Human evaluation

- Relevance : relevant to the ground-truth explanations
- Coherence : logical and fluent
- Informativeness : contains specific information, instead of vague descriptions only



Select The Best Generated Explanation

Please check the definitions before selecting the best explanation:

- **Relevance:** details in the generated explanation are consistent and relevant to the ground-truth explanation's.
- **Coherent:** sentences in the generated explanation are logical and fluent.
- **Informativeness:** generated explanation contains specific information, instead of vauge descriptions only.

Explanations:

Ground Truth Explanation

Best theater ever. Great seats great service. You gonna spend some money but it's worth it if your a movie buff. Got to go

Generated Explanation 1

Great food! Great atmosphere! The seats are very comfortable.

Generated Explanation 2

food great food seats !

Generated Explanation 3

Great food. Great seats, excellent food and good drinks. A great service!

Generated Explanation 4

great great

Questions:

Which one is the most <u>relevant</u> explanation ?

○ Explanation 1 ○ Explanation 2 ○ Explanation 3 ○ Explanation 4

Which one is the most <u>coherent</u> explanation ?

○ Explanation 1 ○ Explanation 2 ○ Explanation 3 ○ Explanation 4

Which one is the most <u>informative</u> explanation ?

○ Explanation 1 ○ Explanation 2 ○ Explanation 3 ○ Explanation 4

Submit

Conclusion

- They unify aspect planning and lexical constraints.
- Compared to existing methods, the quality of the generated explanations is improving.